UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/226,939	01/08/1999	JOHN K. VINCENT	063170.6289	8916

5073 7590 12/31/2007

BAKER BOTTS L.L.P.
2001 ROSS AVENUE
SUITE 600
DALLAS, TX 75201-2980

EXAMINER	
LY, ANH	

ART UNIT	PAPER NUMBER
2162	

NOTIFICATION DATE	DELIVERY MODE
12/31/2007	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptomail1@bakerbotts.com
glenda.orrantia@bakerbotts.com

Office Action Summary	Application No.	Applicant(s)	
	09/226,939	VINCENT ET AL.	
	Examiner	Art Unit	
	Anh Ly	2162	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 October 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *40-73* is/are pending in the application.

4a) Of the above claim(s) *1-39* is/are withdrawn from consideration.

5)☒ Claim(s) *52-59, 64 and 69* is/are allowed.

6)☒ Claim(s) *40-51,60-63,65-68 and 70-73* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is response to Applicants' AMENDMENT and RCE filed on 10/18/2007.

Request for Continued Examination (RCE)

2. The request filed on 10/18/2007 for a Request for Continued Examination (RCE) under 37 CFR 1.114 based on parent Application No. 09/226,939 is acceptable and a RCE has been established. An action on the RCE follows.

3. Claims 1-39 were cancelled.

4. Claims 70-73 have been added.

5. Claims 40-73 are pending in this Application.

Response to Arguments

6. Applicant's arguments, see pages 15-17 in the Remarks, filed 10/18/2007, with respect to the rejection(s) of claim(s) 40, 60 and 65 under "recursively querying a database for one or more dependencies of procedural code ..." have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of WO 92/15066 (PCT/US92/01458) issued to Ming-Chien SHAN dated 02/26/1991.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 60-64 rejected under 35 U.S.C. 101 because the system is not specify or missing a piece of physical object to implement the steps or functions in the claimed invention. So, it is software per se.

"Software per se" is non-statutory under 35 USC 101 because it is merely a set of instructions without any defined tangible output or tangible result being produced. The requirement for tangible result under 35 USC 101 is defined in *State Street Bank & Trust Co. v. Signature Financial Group Inc.*, 149 F.3d 1368, 47USPQ2d 1596 (Fed. Cir. 1998).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to

consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 40-42, 46-51, 60-62 and 65-67 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No.: US 5,586,328 issued to Caron et al. (hereinafter Caron) in view of US Patent No.: US 6,199,063 B1 issued to Colby et al. (hereinafter Colby), and further in view of WO 92/15066 (PCT/US92/01458) issued to Ming-Chien SHAN dated 02/26/1991 (hereinafter SHAN).

With respect to claim 40, Caron a method of generating dependency information for code objects stored in a database (figs. 4 and 8, col. 10, lines 25-32); comprising:

dependencies of procedural code objects stored in the database (the layout dependency code is stored on the storage device or in main memory item 38 in fig. 1: col. 6, lines 30-38 and col. 4, lines 48-50); and

generating a dependency information tracking array based on the indication of one or more dependencies of procedural code objects (figs. 4 and 8, creating a code dependency by a procedure call: col. 10, lines 25-32).

Caron teaches generating dependency code layout for the object code from a program storing on the main memory. Caron further teaches recursive method for generating dependency code (see fig. 12 and col. 15, lines 40-45). Caron does not explicitly teach querying a database and identifying code objects stored in the database.

However, Colby teaches using SQL statement or command to query a relational database (fig. 6, col. 3, lines 10-25, col. 7, lines 60-67 and col. 8, lines 1-35); identifying the database objects (col. 8, lines 5-35 and col. 15, lines 30-50).

Therefore, based on Caron in view of Colby, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the teachings of Caron with the teachings of Colby to utilize the use of SQL statement to query on the database and identifying the objects in the database as disclosed (Colby's figs. 6, and col. 8, lines 1-35), into the system of Caron for the purpose of implementing queuing and recursion the data stored in a database, thereby, enabling a user to have a quickly and efficiently deriving answer to a relational database query (Colby's col. 1, lines 20-22 and col. 3, lines 65-67). In combination of Caron and Colby do not teach explicitly recursively querying a database using a SQL statement.

However, SHAN teach recursive query with SQL statements to get the requested information (page 1, the last paragraph, and pages 9-12 under recursive queries section).

Therefore, based on Caron in view of Colby, and further in view of SHAN, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teachings of SHAN to the system of Caron to have recursively a SQL statement to query a database with objects stored in the relational database with codes stored in the memory. One having ordinary skill in the art would have found it motivated to utilize the use of recursive query with SQL statement on a database to obtain the requested information from a database as disclosed (SHAN's pages 9-12), into the system of Caron for the purpose of evaluating a recursive query of a database (SHAN's page 1), thereby, enabling the user to translate a query in a form for efficient evaluation in large and complex database systems (SHAN's page 11).

With respect to claim 41-42, Caron a method of generating dependency information for code objects stored in a database (figs. 4 and 8, col. 10, lines 25-32); comprising:

dependencies of procedural code objects stored in the database (the layout dependency code is stored on the storage device or in main memory item 38 in fig. 1: col. 6, lines 30-38 and col. 4, lines 48-50); and

generating a dependency information tracking array based on the indication of one or more dependencies of procedural code objects and incorporating the one or more dependencies of specifications of object-oriented code objects into the dependency information tracking array (figs. 4 and 8, creating a code dependency by a procedure call: col. 10, lines 25-32).

Caron teaches generating dependency code layout for the object code from a program storing on the main memory. Caron further teaches recursive method for generating dependency code (see fig. 12 and col. 15, lines 40-45). Caron does not explicitly teach querying a database and identifying code objects stored in the database.

However, Colby teaches using SQL statement or command to query a relational database (fig. 6, col. 3, lines 10-25, col. 7, lines 60-67 and col. 8, lines 1-35); identifying the database objects (col. 8, lines 5-35 and col. 15, lines 30-50).

Therefore, based on Caron in view of Colby, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the teachings of Caron with the teachings of Colby to utilize the use of SQL statement to query on the database and identifying the objects in the database as disclosed (Colby's

figs. 6, and col. 8, lines 1-35), into the system of Caron for the purpose of implementing queuing and recursion the data stored in a database, thereby, enabling a user to have a quickly and efficiently deriving answer to a relational database query (Colby's col. 1, lines 20-22 and col. 3, lines 65-67). In combination of Caron and Colby do not teach explicitly recursively querying a database using a SQL statement.

However, SHAN teach recursive query with SQL statements to get the requested information (page 1, the last paragraph, and pages 9-12 under recursive queries section).

Therefore, based on Caron in view of Colby, and further in view of SHAN, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teachings of SHAN to the system of Caron to have recursively a SQL statement to query a database with objects stored in the relational database with codes stored in the memory. One having ordinary skill in the art would have found it motivated to utilize the use of recursive query with SQL statement on a database to obtain the requested information from a database as disclosed (SHAN's pages 9-12), into the system of Caron for the purpose of evaluating a recursive query of a database (SHAN's page 1), thereby, enabling the user to translate a query in a form for efficient evaluation in large and complex database systems (SHAN's page 11).

With respect to claim 46, Caron teaches further comprising compiling one or more code objects stored in the database in debug mode using a database code object debugging tool (processing of debugging source code to a compiled program: col. 1, lines 55-60).

With respect to claim 47, Caron in view of Colby teaches further comprising identifying one or more dependent objects stored in the database that are INVALID (Colby's fig. 19, col. 21, lines 1-18).

With respect to claim 48, Colby teaches further comprising identifying one or more cyclic dependencies among code objects stored in the database (col. 7, lines 4-15 and col. 10, lines 18-32).

With respect to claim 49, Colby teaches wherein identifying one or more cyclic dependencies comprises utilizing a graph traversal algorithm to identify one or more cyclic dependencies (col. 7, lines 4-15 and col. 10, lines 18-32).

With respect to claim 50, Colby teaches further comprising generating a dependency graph for code objects stored in the database based at least in part on the dependency information tracking array (col. 7, lines 4-15 and col. 10, lines 18-32).

Claim 60 is essentially the same as claim 40 except that it is directed to a system rather than a method, and is rejected for the same reason as applied to the claim 40 hereinabove.

Claim 61 is essentially the same as claim 41 except that it is directed to a system rather than a method, and is rejected for the same reason as applied to the claim 41 hereinabove.

Claim 62 is essentially the same as claim 42 except that it is directed to a system rather than a method, and is rejected for the same reason as applied to the claim 42 hereinabove.

Claim 65 is essentially the same as claim 40 except that it is directed to a computer readable medium rather than a method, and is rejected for the same reason as applied to the claim 40 hereinabove.

Claim 66 is essentially the same as claim 41 except that it is directed to a computer readable medium rather than a method, and is rejected for the same reason as applied to the claim 41 hereinabove.

Claim 67 is essentially the same as claim 42 except that it is directed to a computer readable medium rather than a method, and is rejected for the same reason as applied to the claim 42 hereinabove.

9. Claims 43-45, 51, 63 and 68 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No.: US 5,586,328 issued to Caron et al. (hereinafter Caron) in view of US Patent No.: US 6,199,063 B1 issued to Colby et al. (hereinafter Colby) and further in view of WO 92/15066 (PCT/US92/01458) issued to Ming-Chien SHAN dated 02/26/1991 (hereinafter SHAN) and in view of US Patent No. 5,926,819 issued to Doo et al. (hereinafter Doo).

With respect to claim 43, Caron in view of Colby and SHAN discloses a method of generating dependency information as discussed in claim 40.

Caron, Colby and SHAN disclose substantially the invention as claimed.

Caron, Colby and SHAN do not teach parsing the source code of the database for data manipulation statements that fire triggers; and identifying one or more data manipulation statements that fire triggers.

However, Doo teaches DML statement being applied to fire the triggers (col. 5, lines 22-42, also see figs, 2-4) and parsing the tree of DML statement (col. 5, lines 38-42 and col. 6, lines 50-58).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made for modifying the teachings of Caron in view of Colby and SHAN with the teachings of Doo by incorporating the use of data manipulation statements such as DML to fire the triggers. The motivation being for having secure triggers in order to prevent the replication or even corrupt other data in the database (Doo's col. 2, lines 58-62).

With respect to claim 44, Caron in view of Colby and SHAN discloses a method of generating dependency information as discussed in claim 40.

Caron, Colby and SHAN disclose substantially the invention as claimed.

Caron, Colby and SHAN do not teach parsing the source code for UPDATE, DELETE or INSERT statements.

However, Doo teaches DML statements being applied to INSERT, DELETE and UPDATE operations (col. 5, lines 5-12).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made for modifying the teachings of Caron in view of Colby and SHAN with the teachings of Doo by incorporating the use of data manipulation

statements such as DML to fire the triggers. The motivation being for having secure triggers in order to prevent the replication or even corrupt other data in the database (Doo's col. 2, lines 58-62).

With respect to claim 45, Caron in view of Colby teaches a method of generating dependency information as discussed in claim 40.

Caron teaches generating dependency code layout for the object code from a program storing on the main memory. Caron further teaches recursive method for generating dependency code (see fig. 12 and col. 15, lines 40-45). Colby teaches using SQL statement or command to query a relational database and identifying the database objects.

Therefore, based on Caron in view of Colby, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the teachings of Caron with the teachings of Colby to utilize the use of SQL statement to query on the database and identifying the objects in the database as disclosed (Colby's figs. 6, and col. 8, lines 1-35), into the system of Caron for the purpose of implementing queuing and recursion the data stored in a database, thereby, enabling a user to have a quickly and efficiently deriving answer to a relational database query (Colby's col. 1, lines 20-22 and col. 3, lines 65-67). In combination of Caron and Colby do not teach explicitly recursively querying a database using a SQL statement.

However, SHAN teach recursive query with SQL statements to get the requested information (page 1, the last paragraph, and pages 9-12 under recursive queries section).

Therefore, based on Caron in view of Colby, and further in view of SHAN, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify the teachings of SHAN to the system of Caron to have recursively a SQL statement to query a database with objects stored in the relational database with codes stored in the memory. One having ordinary skill in the art would have found it motivated to utilize the use of recursive query with SQL statement on a database to obtain the requested information from a database as disclosed (SHAN's pages 9-12), into the system of Caron for the purpose of evaluating a recursive query of a database (SHAN's page 1), thereby, enabling the user to translate a query in a form for efficient evaluation in large and complex database systems (SHAN's page 11).

With respect to claim 51, Caron in view of Colby and SHAN discloses a method of generating dependency information as discussed in claim 40.

Caron, Colby and SHAN disclose substantially the invention as claimed.

Caron, Colby and SHAN do not teach database catalog.

However, Doo teaches the database system stores information as metadata in a data dictionary within the database (col. 6, lines 27-32).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made for modifying the teachings of Caron in view of Colby and SHAN with the teachings of Doo by incorporating the use of data manipulation statements such as DML to fire the triggers. The motivation being for having secure triggers in order to prevent the replication or even corrupt other data in the database (Doo's col. 2, lines 58-62).

Claim 63 is essentially the same as claim 43 except that it is directed to a system rather than a method, and is rejected for the same reason as applied to the claim 43 hereinabove.

Claim 68 is essentially the same as claim 43 except that it is directed to a computer-readable medium rather than a method, and is rejected for the same reason as applied to the claim 43 hereinabove.

10. Claims 70-73 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No.: US 5,586,328 issued to Caron et al. (hereinafter Caron) in view of US Patent No.: US 6,199,063 B1 issued to Colby et al. (hereinafter Colby) and further in view of WO 92/15066 (PCT/US92/01458) issued to Ming-Chien SHAN dated 02/26/1991 (hereinafter SHAN) and in view of US Patent No. 5,805,804 issued to Laursen et al. (hereinafter Laursen).

With respect to claims 70-71, Caron in view of Colby and SHAN discloses a method of generating dependency information as discussed in claim 40.

Caron, Colby and SHAN disclose substantially the invention as claimed. Caron, Colby and SHAN do not teach PL/SQL specifications for a collection of stored functions and procedures identified as a single entity.

However, Laursen teaches PL/SQL statements (detx 59)

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made for modifying the teachings of Caron in view of Colby and SHAN with the teachings of Laursen by incorporating the use of PL/SQL for

collecting of stored functions. The motivation being for selecting, collecting, retrieving and delivering information over the network in real-time or non-real-time (Laursen's col. 1, lines 10-10 and col. 2, lines 1-12).

With respect to claims 72-73, Caron in view of Colby and SHAN discloses a method of generating dependency information as discussed in claim 40.

Caron, Colby and SHAN disclose substantially the invention as claimed. Caron, Colby and SHAN do not teach one of the one or more dependencies already occurs in the graph and displaying a dependency graph to a user, the dependency graph generated based at least in part on the dependency information tracking array..

However, Laursen teaches directed graph and tracking arrays (dext 99,101,121,125).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made for modifying the teachings of Caron in view of Colby and SHAN with the teachings of Laursen by incorporating the use of PL/SQL for collecting of stored functions. The motivation being for selecting, collecting, retrieving and delivering information over the network in real-time or non-real-time (Laursen's col. 1, lines 10-10 and col. 2, lines 1-12).

Allowable Subject Matter

11. The following is a statement of reasons for the indication of allowable subject matter: Claims 52-59, 64 and 69 are allowed because these claims have some distinct

features: "parsing the source code of the database for data manipulation statements that fire triggers and generate a dependency information tracking array based on the indications of one or more dependencies of procedural code objects stored in the database, one or more dependencies of specifications of object-oriented code objects stored in the database, one or more dependencies of implementations of object-oriented code objects stored in the database and one or more dependencies on triggers of code objects stored in the database." These distinct features, in conjunction with all other limitations of the dependents and independent claims render claims 52-59, 64 and 69 them allowable. If they are overcome the 101 rejection.

12. Claims 52-59, 64 and 69 are allowed.

Contact Information

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ANH LY, whose telephone number is (571) 272-4039 or via e-mail: ANH.LY@USPTO.GOV (written authorization being given by Applicant(s) - MPEP 502.03 [R-2]) or fax to (571) 273-4039 (unofficial fax number direct to examiner's office). The examiner can normally be reached on TUESDAY – THURSDAY from 8:30 AM – 3:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **John Breene,** can be reached on **(571) 272-4107.**

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see **http://pair-direct.uspto.gov**. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). Any response to this action should be mailed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231, or faxed to: Central Fax Center **(571) 273-8300**

Cam Y Truong
Primary Examiner

ANH LY
DEC. 14th, 2007